10 july 1985

To the Comal Standardisation Group

Proposals for the Comal Standard

In continuation of the tradition founded prior to the standard meeting in Stockholm the three Danish manufacturers, RC, Unicomal, and DDE, have worked out the following proposals for the Comal standard.

Because of the late appearance of a revised standard the productions referred are those of the standard of April 13-14, 1984, with corrections of September 1984.

1. To the semantics for the ORD functions (Appendix A, 14) add:

   The ORD function taken of the empty string results in a runtime error.

2. Define the multiplying operators DIV and MOD, production 72, as proposed by Peter Burkinshaw in Stockholm:

   The DIV function is defined to take two arguments x and y and to return INT(x/y), the next lowest integer less then or equal to the result of dividing x by y.

   The MOD function also takes two arguments x and y and returns (x - (x DIV y) * y).

   The examples following production 72 and concerning negative y's are changed to

```
  36 MOD(-5) is -4
(-36) MOD(-5) is -1
  35 MOD(-5) is  0
(-35) MOD(-5) is  0
```

3. Two new functions DATE$ and TIME$ are added to the extensions:

   DATE$ returns the date as a string in the format yyyy-mm-dd

   TIME$ returns the time as a string in the format hh:mm:ss

   DATE$ and TIME$ are added to the list of reserved words.

4. As originally proposed by Borge Christensen, add to the extensions:

   <procedure call statement> ::=
   <procedure call> {; <procedure call>}

   <procedure call> ::=
   [EXEC] <procedure identifier> [<parameter list>]

5. Add a PAGE statement to the extensions:

   <page statement> ::= PAGE

   The PAGE statement outputs a form feed character to the current output unit.

6. As supported by the development group in Stockholm add to the extensions two new functions:

   MAXINDEX (<array identifier> [, <numeric expression>])
   MININDEX (<array identifier> [, <numeric expression>])

<array identifier> ::= <numeric identifier> |
<string identifier>

The <array identifer> must be the name of an array. The <numeric expression> evaluates to a legal dimension number, otherwise a runtime error occurs. If <numeric expresson> is omitted the default value 1 is used. The first dimension is numbered 1.

MAXINDEX returns the upper bound of the referred dimension, while MININDEX returns the lower bound of the dimension.

7. As considered in Cambridge three bitwise operators are added to the extensions:

BITAND
BITOR
BITXOR

All three are dyadic operators, resulting in bitwise and/or/xor, respectively, of the operands. The precedence of BITAND, BITOR, and BITXOR are as of the multiplying operators.

8. This proposal deals with a simple extension of <format info> to allow formatting of strings and output of numeric values in exponential format and should be added to the extensions.

Production 132 is changed to

<using element> ::= <numeric expression> |
<string expression>

Add to the semantics:

A numeric field immidiately followed by 3 or more up arrows (^) will be substituted on output by the value , written in exponential format of a corresponding numeric expression. The mantissa will be normalised with possible leading spaces, and the exponent will be written as 'E' 'sign' 'value of exponent with leading zeroes'.

A substring of one or more embedded less than signs (<) will be substituted on output by the value of a corresponding string expression. The value of the string expression is left justified and right truncated if longer than the field.

A substring of one or more embedded greater than signs (>) will be substituted on output by the value of a corresponding string expression. The value of the string expression is right justified and right truncated if longer than the field.

A substring of one or more embedded exclamation mark (!) will be substituted on output by the value of a corresponding string expression. The value of the string expression is centered within the field and right truncated if longer than the field.

The characters #, <, >, !, ^, and ' is output by preceding them by a single quote (').

9. The next proposal is directed to an attempt to define a recommandation for a minimal command set for Comal. You may consider this proposal as a first contribution towards a definition of what may be called a 'Comal Standard Environment'.

We propose here a number of commands, that every Comal

interpreter should respond to, and all in the same way. Some af the commands are only relevant if the interpreter uses line numbers as part of the Comal statements.

The following functions are covered by the commands:

a. Automatic generation of line numbers
b. Return from the Comal system
c. Delete files
d. List catalog
e. Enter Comal program in ascii form
f. List Comal program in ascii form
g. Load Comal program in compressed form
h. Restart the Comal system
i. Remove lines from a program
j. Renumber the lines of a program
k. Execute a program
l. Save Comal program in compressed form
m. Select output unit

As can be seen from the above no functions are defined for editing a program. These functions we prefer to defer to a later stage.

General to a number of the above mentioned functions is the notion of a line range. A line range is defined as follows:

<line range> ::= <first line> [- [<last line>]] ¦
- <last line>

<first line> ::= <integer>

<last line> ::= <integer>

The meaning of a line range is:

xxx	line xxx
xxx -	from line xxx to the end of the program, line xxx inclusive
xxx - yyy	from line xxx to line yyy, lines xxx and yyy inclusive
- yyy	from the beginning of the program to line yyy, line yyy inclusive

Also general to a number of the functions is the notion of a unit name. The unit name is an implementation dependent name of an i/o unit, for example a file or a printer. Unit name is a string constant, and thus surrounded by double quotes (").

The proposed commands, the syntax and semantics are:

a. AUTO [<first line>] [, <increment>]

Automatic generation of line numbers. The line numbers generated are first line, first line + increment, first line + 2 * increment, etc. The default value of <increment> is 10, and the default value of <first line> is the greatest line number used + increment. If no program exists the greatest line number used is 0.

b. BYE

Return from the Comal system.

c. DELETE <unit name>

Deletes the specified unit.

d. DIR [<unit name>]

List the catalog of the specified unit. If <unit name> is omitted the catalog of the current directory is listed.

e. ENTER <unit name>

Enters a program in ascii form, for example written by the LIST command. The ENTER command executes an implicit NEW command before entereing the program.

f. LIST [<line range>] [<unit name>]

The program is written to the specified unit. If <line range> is not given the entire program is listed, and if <unit name> is not given the program is written to the current output unit.

The list appears with indentations for every structure.

An existing file can not be overwritten by the LIST command.

g. LOAD <unit name>

Loads a program in compressed form. The program must be created by a SAVE command. Before loading a program an implicit NEW is executed.

h. NEW

Restart the interpreter.

i. REMOVE <line range>

Removes lines from a program.

j. RENUMBER [<first line>] [, <increment>]

Renumbers the lines of the program. The default value of <first line> is 10, and the default value of <increment> is 10.

k. RUN [<unit name>]

Execute program. If <unit name> is given the function is defined by the command sequence:

LOAD <unit name>
RUN

l. SAVE <unit name>

The program is written to the specified unit in compressed form.

m. SELECT OUTPUT <unit name>

Selects another unit for current output unit. The selected unit remains current output unit until a new SELECT OUTPUT command is executed.

RC,
Unicomal,
DDE.

To the COMAL Standardisation Group

Proposal for the COMAL Standard

10. To the semantics for the FOR-NEXT statement (following production 27) add:

    The control variable is local to the FOR-statement. It is not allowed to assign to the control variable, it is only allowed to refer it. The control variable is undefined after the FOR-NEXT statement.

RC,
UniComal,
DDE.